IN THE SPECIFICATION

Please amend the specification as follows:

In the paragraph on page 8 from lines 13-20, please amend the paragraph as follows:

-- Each CPU 12A-12H may include a segment of cache memory 13A-13H for storage of frequently accessed data and programs. Maintaining coherency among the plurality of caches in the CPUs 12A-12H is important to the efficient operation of the system 10. Maintaining coherency among the caches found in each CPU 12A-12H is further complicated by the split-bus configuration since coherency should be maintained between the separate buses 14A and 14B. Also, because requests may originate from or may be directed to not only one of the CPUs 12A-12H, but also from one of the peripheral devices 32A-32B, 34A-34B, or 36A-36B, cache coherency should be maintained along the I/O bus 27, as well. --

In the paragraph on page 11 from lines 6-15, please amend the paragraph as follows:

-- Turning generally to Fig. 3, a more detailed block diagram of the host controller 16 and internal bus structure 30 is illustrated. As can be seen in Fig. 3, the internal bus structure 30 comprises a plurality of buses, such as buses 30A-30K. The buses 30A-30K and associated signals will be discussed more specifically below. For each request or cycle that is delivered to the host controller 16, there are a number of signals and internal responses that are delivered via individual buses 30A-30K in the present system. Thus, for each request there are a series of ordered exchanges 38A-38K of various signal types among the various modules within the host controller 16. The specific exchanges 38A-38K on each bus 30A-30K associated with each request or cycle are design dependent. However, general techniques discussed below may be implemented to provide a standard design for the internal bus structure 30 in which specific implementations may be achieved. --

In the paragraph beginning on page 11, line 17 and ending on page 12, line 11, please amend the paragraph as follows:

-- One aspect of the internal bus structure 30 is that each transaction associated with a particular request is exchanged on a single unidirectional bus 30A-30K. As previously discussed when a request is received by a processor controller PCON, a series of transactions or exchanges 38A-38K among the various components in the system is conducted. Not until the plurality of transactions 38A-38K associated with each request are completed can the request ultimately be retired. Present internal bus structure 30 conducts each transaction or exchange 38A-38K via an associated signal on a separate unidirectional bus 30A-30K. By executing each transaction 38A-38K associated with a request on a separate, individual bus 30A-30K, the routing congestion and design complexity are minimized. Because each bus 30A-30K is unidirectional and since each bus 30A-30K has an associated signal (i.e., each bus 30A-30K has a unique signal associated with it and signals are not shared between buses 30A-30K), certain

connections in the system can be eliminated. For instance, the bus 30B in the present embodiment carries an Initial Response signal. As will be further explained below, the Initial Response signal is delivered only from the tag controller TCON to an associated processor controller, here PCON0. Thus, rather than providing a signal path (i.e., an additional bus) to the memory controller MCON, the bus associated with the Initial Response signal, here bus 30B, is only connected between the tag controller TCON and the processor controller PCON0. Thus, a bus connection to the memory controller MCON may be eliminated thereby reducing routing congestion. --

In the paragraph beginning on page 13, line 19 and ending on page 14, line 3, please amend the paragraph as follows:

-- Further, because the present system centralizes coherency control and cycle ordering in a single module, here the tag controller TCON, the point-to-point connections allow for a multiplicity of processor and I/O bus segments to be connected or removed from the system easily without consideration of coherency and cycle ordering. The internal bus structure protocol avoids direct processor-to-processor and processor-to-I/O communication by routing information through central modules which facilitate exchanges 38A-38K with each processor and I/O interface. This allows a variable number of new processor and I/O modules to be connected without having to re-design or alter the existing system modules. --

In the paragraph beginning on page 14, line 19 and ending on page 15, line 15, please amend the paragraph as follows:

-- Another advantage of the present system is the identification (ID) tagging structure. As a processor controller PCON receives a request from an agent on a corresponding bus, the request is tagged by the processor controller PCON such that it includes a unique ID, such as an identification tag 42, containing source, destination, and cycle information. The source/cycle/destination ID is carried through each transaction of each request. The source identification (ID) 44 corresponds to the source of the request (e.g., PCON0-PCON2 or TCON). The destination ID 46 corresponds to the destination of the request (e.g., PCON0-PCON2, MCON, TCON, Config., broadcast). The cycle ID 48 corresponds to a unique ID for each request and may include a READ/WRITE bit to identify the request type and a toggle bit to be discussed further below. By retaining the source/cycle/destination ID through each transaction issued on the individual buses 30A-30K of the internal bus structure 30 for a corresponding request, each transaction 38A-38K can be completed out of order since the particular transactions 38A-38K of specific signal types can be easily tracked. In prior systems, because ID information was not retained for each of the transactions ~~transaction~~ 38A-38K on a particular request, the transactions 38A-38K had to be executed in a particular sequence so that the system could identify each of the transactions ~~transaction~~ 38A-38K and associate it with the current request being executed. By allowing for out of order returns of transactions 38A-38K on the buses 30A-30K, system performance can be increased. Further, since source and destination identification data is carried through each of the transactions ~~transaction~~ 38A-38K for every request, adding other interfaces, such as a fourth processor controller PCON3 (not shown), is simplified. The processor controller PCON3 would have a unique identification to provide

source and destination ~~ID~~ IDs 44 and 46 for any requests being processed through the processor controller PCON3. --

In the paragraph beginning on page 15, line 17 and ending on page 16, line 8, please amend the paragraph as follows:

-- As previously discussed, a toggle bit 50 may be added to the cycle ID 48 to allow an ID to be used after a request is effectively completed, but before each and every transaction 38A-38K for the request has been received. For example, the processor controller PCON may receive a request and initiate the request to the corresponding module such as the memory controller MCON. Once the processor controller PCON initiates the request through the memory controller MCON or the tag controller TCON, the toggle bit 50 can be set and the cycle ID 48 can be reused in the queue by the processor controller PCON since the processor controller PCON has completed its task. However, at this point, the request may not be retired since the other transactions 38A-38K corresponding to the memory controller MCON and the tag controller TCON have not been completed. The processor controller PCON, on the other hand, is finished with its portion of the request. The toggle bit 50 is provided such that it clears a location in the associated queue for another transaction 38A-38K. By toggling the toggle bit 50, the rest of the system sees the queue location as a new address and can thereby deliver another request to the queue in the processor controller PCON before the original request is retired. This mechanism effectively doubles the number of unique transactions 38A-38K without having to double the size of the request queues. --

In the paragraph on page 17 from lines 4-12, please amend the paragraph as follows:

-- Referring more specifically to Fig. 3, an exemplary request and associated transactions are described. An exemplary embodiment utilizing specific signals and transactions 38A-38K on the buses 30A-30K of the internal bus structure 30 is illustrated. The specific request transactions and buses described herein are exemplary and are used for one particular implementation. However, the techniques discussed above corresponding to the protocol and design techniques for the internal bus structure 30 are applicable to other designs implementing various other signals corresponding to requests. Further, while buses 30A-30K are illustrated and described with specific reference to the processor controller PCON0, similar buses are provided for the remaining modules, such as processor controllers PCON1 and PCON2. --